Energy & Buildings 278 (2023) 112635

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb

A knowledge embedded graph neural network-based cooling load prediction method using dynamic data association

Zhiwen Chen [a,b,*], Zhengrun Zhao [a], Qiao Deng [a], Peng Tang [b,*], Chunhua Yang [a], Xinhong Li [c], Weihua Gui [a,b]

[a] School of Automation, Central South University, Changsha 410083, China
[b] Pengcheng Laboratory, Shenzhen 518055, China
[c] Shenzhen DAS Intelligent Technology Ltd. Co., Shenzhen 518057, China

ARTICLE INFO

Article history:
Received 13 July 2022
Revised 15 October 2022
Accepted 2 November 2022
Available online 9 November 2022

Keywords:
Building energy saving
Cooling load prediction
Dynamic data association
Graph neural network
Knowledge embedding

ABSTRACT

Accurate cooling load prediction has continually attracted interest due to its important role in building energy saving. Much attention to existing methods has been paid to the temporal information of the features that affect the load prediction. However, facing the time-varying interactions between various types of features, existing approaches have limited ability to extract the implicit information of the complex associations among them. This limitation leads to underutilization of features, especially those that indirectly affect the load prediction through such interactions, and further constrain prediction accuracy. To this end, this paper proposed a graph neural network-based cooling load prediction method to integrate the associations and temporal information of the features. Using the weighted mask mechanism, a knowledge embedded dynamic association graph is constructed to handle the time-varying feature associations rationally. By introducing empirical knowledge and adjusting the graph structure automatically according to the real-time data, the proposed method can provide accurate cooling load prediction. Comprehensive experiments are carried out on the onsite data, and the experimental results show the practicability and effectiveness of the proposed method for cooling load prediction.

1. Introduction

Building is one of the major energy consumers, accounting for 40% of global annual energy consumption [1]. To maintain the indoor thermal comfort, the heating, ventilation and air conditioning (HVAC) systems have been widely used in buildings, and such systems consume about 50% of the total energy consumption of buildings [2]. Accurate cooling load (*CL*) prediction can guide the management and operation of the HVAC systems and optimize the energy utilization [3], which is essential in the building energy saving. In this regard, various studies have been carried out to suggest accurate *CL* prediction methods. In the earlier research, the calculation of *CL* was mostly based on the estimation model, such as transfer function method (TFM) [4], monthly/ seasonal method [5], the radiant time series method [6] and so on. However, to achieve satisfactory results by using these estimation models, professional knowledge and architectural information are often required, which need experienced operators and are hard to be obtained in practice, respectively.

With more and more buildings equipped with a significant number of sensors to collect real-time data, data-driven prediction methods have been widely used in *CL* prediction. Deng [7] et al. applied multiplicative decomposition and Autoregressive Integrated Moving Average to Singapore power data. A robust hourly *CL* forecasting method based on time-indexed autoregressive with exogenous inputs models is introduced in [8]. Vaghefi [9] et al. combined linear regression (LR) and seasonal autoregressive moving average model for cooling and electricity load prediction. As a classical data-driven method, support vector machine (SVM) has been widely used in the field of prediction [10]. Fan [11] et al. proposed a *CL* prediction method whose input variables of SVM are calibrated with Monte-Carlo simulations and stochastic treatment offline. Idowu [12] et al. compared multiple machine learning methods and found that SVM gave the best prediction performance. Besides SVM, considerable attention has been focused on artificial neural networks (ANN) because of their strong generality. Ben-Nakh [13] et al. used general regression neural network to obtain relatively ideal *CL* profiles prediction results. Radial basis function neural network is used to predict the energy consumption of fan-coil systems [14] and air-conditioning load [15]. Wang [16] et al. proposed an ANN and an ensemble approach, which exhibited a better load forecasting effect by combining complementary

* Corresponding authors.
E-mail addresses: zhiwen.chen@csu.edu.cn (Z. Chen), zhaozr@csu.edu.cn (Z. Zhao), qiao.deng@csu.edu.cn (Q. Deng), tangp@pcl.ac.cn (P. Tang), ychh@csu.edu.cn (C. Yang), lixh@chn-das.com (X. Li), gwh@csu.edu.cn (W. Gui).

https://doi.org/10.1016/j.enbuild.2022.112635
0378-7788/

sub-models. Recently, deep learning has gained increasing popularity in the *CL* prediction due to the strong fitting ability and flexibility in model development. Among these methods, the recurrent neural network (RNN) was illustrated as a promising way for the *CL* prediction [17]. Sendra-Arranz [18] et al. used three different forms of long short term memory neural network (LSTM) to predict the energy consumption of HVAC systems in a self-sufficient solar house. Li [19] et al. proposed a neural network architecture with an attention mechanism for developing RNN-based building energy prediction and studied its effectiveness in explaining *CL* prediction. Wang [20] et al. developed a variety of models and conducted detailed experiments on long-term and short-term prediction, and the results show that LSTM has advantages in short-term prediction and is more robust to weather forecast uncertainly.

Due to the strong temporal correlation of the features that affect the *CL* prediction, most studies mentioned above mainly focus on temporal feature extraction but rarely analyze the interdependency among the features. In fact, changes in some features may cause fluctuations in the others, and those that indirectly affect load prediction can be used more rationally when considering the associations among the features. Besides, many studies pay more attention to meteorological data which affects the load prediction more directly [21]. But the chiller plant operation data, which are informative and related to *CL* prediction, has been underestimated.

In order to address the problems mentioned above, a suitable method is needed to characterize and utilize the associations among the features. A large number of available onsite data makes the data association of features easier to be obtained, and more features can be used conveniently in addition to meteorological information. However, traditional *CL* prediction methods usually treat the features as independent or only temporal correlated. The graph convolutional neural network (GCN) [22] has been a promising tool to deal with such interacted features due to its excellent ability to deal with graph structure data, which are represented by nodes and edges. As one of the representative models which combine the temporal information extraction methods with GCN, the spatio-temporal graph convolutional neural network (STGCN) [23] achieves great prediction effectiveness in the time-series prediction task. However, in the *CL* prediction, this graph neural network-based method still has limitations:

(1) The interactions among features are time-varying, and the fixed graph structure of STGCN fails to reflect the dynamic associations among the features.
(2) The features with various physical meanings have different effects on the *CL* prediction. The graph structure of STGCN has limited ability to quantify such meanings, which would therefore be fuzzified because of the information aggregation operation of the GCN, thereby reducing the diversity of information.

Therefore, a novel graph neural network-based cooling load prediction method is proposed in this paper, named knowledge embedded associative-temporal dynamic graph convolutional network (K-ATDGN). Considering the temporal correlation of the features and the interactions among them, this method uses GCN and gated CNN as the associative module and temporal module, to integrate the association information and the temporal domain information. With the aim of solving the problem that a fixed graph cannot describe the time-varying associations, a dynamic and informative association graph is constructed according to the historical and real-time data. Additionally, to maintain the diversity of information, this paper proposes the classification guidelines for features, based on which a knowledge embedded weighted mask mechanism is further proposed as a knowledge embedding method to make the association graph informative. The knowledge embedded graph contributes to providing more implicit information for the *CL* prediction. Finally, the proposed method is verified by onsite data, and shows superior prediction performance. The contributions of this paper are summarized as follows:

(1) **A novel cooling load prediction method is proposed, which integrates the dynamic associations and temporal information of data.** The proposed method uses the dynamic association graph to quantify the associations among the features that affect the *CL* prediction, which can dynamically adjust the graph structure according to the real-time onsite data. Combined with the temporal information, the proposed method can predict the *CL* accurately.
(2) **A knowledge embedding method for the association graph is proposed.** By using a knowledge embedded weighted mask to reasonably adjust the associations among features, the refined dynamic association graph can guide a more effective information aggregation, further improving the prediction performance.
(3) **Comprehensive comparative experiments are carried out on the onsite data.** The proposed K-ATDGN is compared with several *CL* prediction methods and shows superior prediction performance and practical value.

The rest of this paper is arranged as follows: Section 2 introduces the features that affect the *CL* prediction and the composition of the chiller plant. Section 3 describes the construction method of knowledge embedded dynamic association graph, and the structure of K-ATDGN. In Section 4, the effectiveness of the proposed association graph and the practicability of the K-ATDGN in the *CL* prediction are verified through comprehensive comparative experiments. Section 5 gives the conclusion and the future work.

2. Preliminary

2.1. Features that affect cooling load prediction

As shown in Fig. 1, there are many features can affect the *CL* prediction:

(a) Meteorological features: The heat absorption and release of buildings depend on meteorological conditions, which directly affect the *CL* prediction. For example, the outdoor temperature is associated with the heat absorption of the building, which brings a change in the *CL*.

(b) Time features: The *CL* is various from time to time. Many features are associated with time, such as solar radiation and outdoor temperature changes on a daily basis and personal occupancy changes on a weekly basis, significantly impacting the *CL* prediction.

(c) Equipment operation features: The operating status of the chiller plant can provide lots of information for *CL* prediction. For example, the flow rate and temperature difference of chilled water directly affect the indoor thermal balance, which are related to the frequency of the pumps and the operating status of chillers, respec-

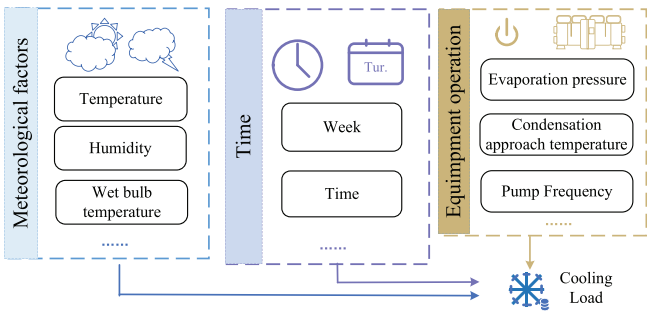

Fig. 1. Features that affect the cooling load prediction.

tively. To analyze the *CL* more comprehensively, it is necessary to consider the operation of key refrigeration equipment.

As mentioned above, there are many interrelated features. The associations among them can provide additional information for accurate *CL* prediction.

2.2. Chiller plant

The chiller plant is the core of the HVAC system, which carries the majority of the *CL*. As shown in Fig. 2, the chiller plant mainly consists of chillers, chilled water pumps, cooling water pumps, and cooling towers. The equipment works cooperatively to supply cooling for the building. A complete refrigeration process can be regarded as a refrigeration circulation, which can be divided into three sub-circulations: the chilled water circulation, the refrigerant circulation, and the cooling water circulation.

The chilled water circulation takes chilled water as the medium that connects the HVAC cooling terminals and the chillers. The high-temperature inlet chilled water releases heat in the chillers and turns into low-temperature outlet chilled water, which is transported to the designated position to participate in the heat exchange in the terminals. After the heat absorption, high-temperature chilled water is recycled again into the chillers, and a complete chilled water circulation is formed then. The evaporation approach temperatures reflect the heat exchanges between chilled water and chillers, which are essential equipment operation features. The cooling water circulation is responsible for exchanging heat between the outdoor and the chillers. The low-temperature inlet cooling water absorbs heat in the condensers of chillers and is pumped into the outdoor cooling towers. In the towers, the cooling water dissipates heat to the outside with temperature decreases and then flows back to the condensers for the next circulation. The condensation approach temperatures can reflect the heat exchanges between cooling water and chillers.

The chillers are the core equipment of the chiller plant, in which the refrigerant circulation is carried out accompanied by energy consumption. The heat of the high-temperature chilled water is transferred to the refrigerant in the evaporator. The compressor then converts the refrigerant into high-temperature and high-pressure gas through compression. Then, the refrigerant exchanges heat with the cooling water in the condenser, releases heat and changes into a liquid state. The liquid refrigerant flows to the evaporator through the expansion mechanism, during which the temperature and pressure decrease. The condensation pressure and evaporation pressure are related to the phase changes of refrigerant and are closely related to the cooling. It can be seen that each sub-circulation of the refrigeration is indispensable.

3. Proposed method

The K-ATDGN is proposed in this section. In subSection 3.1, the basic structure of the associative-temporal graph convolutional network (ATGN) is introduced, which serves as the basic model for the *CL* prediction. SubSection 3.2 clarifies the construction method of the knowledge embedded dynamic association graph. In subSection 3.3, the framework of K-ATDGN is described.

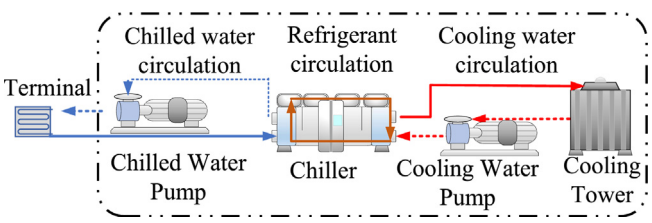

Fig. 2. The schematic diagram of a typical chiller plant.

3.1. ATGN

ATGN follows the idea of STGCN and appropriately simplifies the number of temporal and spatial modules (associative modules). Fig. 3 shows the structure of the ATGN.

(a) **Temporal module**

This paper uses gated CNN as the temporal module for feature extraction in the temporal domain. As shown in Fig. 3, the input of temporal module is $X_\tau \in \mathbb{R}^{C_\tau \times L_\tau \times n}$, where C_τ, L_τ and n represent the number of input channels, number of samples and number of features, respectively. CNNi $(i = a, b)$ can extract temporal information by convolution along the temporal dimension of data, whose convolution kernel is $\Gamma_i \in \mathbb{R}^{K_\tau \times C_\tau \times C\prime_\tau}$. After the convolution, $X_i \in \mathbb{R}^{C\prime_\tau \times (L_\tau - K_\tau + 1) \times n}$ is obtained. The K_τ represents the kernel size, and $C\prime_\tau$ is the number of output channels.

$$Y = (X_a + X\prime_\tau) \odot sigmoid(X_b) \in \mathbb{R}^{C\prime_\tau \times (L_\tau - K_\tau + 1) \times n} \quad (1)$$

The output Y of the temporal module is obtained according to Eq. (1), where $\odot$ represents Hadamard product. To keep more original information of X_τ, the alignment operation is adopted to get $X\prime_\tau \in \mathbb{R}^{C\prime_\tau \times (L_\tau - K_\tau + 1) \times n}$ from X_τ.

(b) **Associative module**

The associative module uses GCN to aggregate the information contained in the features according to the association graph. As shown in Fig. 3, the input of this module includes the node data $X^l \in \mathbb{R}^{C_g \times L_g \times n}$ and the adjacency matrix $A \in \mathbb{R}^{n \times n}$, where n represents the number of nodes in the graph, which is the same as the number of the features. C_g and L_g represent the number of channels and number of samples, respectively. The adjacency matrix element A_{ij} is the weight of the edge between the ith and jth node, which determines the association between them.

$$X^{l+1} = \sigma\left(\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}X^l W\right) \quad (2)$$

Eq. (2) is the forward propagation rule of one-layer graph convolution, where $\tilde{A} = A + I_n, \tilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$. I_n is the identity matrix of size n. Parameter $W \in \mathbb{R}^{C_g \times C\prime_g}$ is trainable, where $C\prime_g$ is the number of output channels, and σ is the activation function.

The input of ATGN is $X_t = (\tilde{x}_t, \tilde{x}_{t-1}, \cdots \tilde{x}_{t-L+1}) \in \mathbb{R}^{1 \times L \times n}$, where $\tilde{x}_t \in \mathbb{R}^{1 \times n}$ represents the sample collected at time instance t, and L represents the number of samples in X_t. ATGN integrates temporal and association information through the temporal and associative modules, and the final output is extracted through a fully connected layer.

3.2. Knowledge embedded dynamic association graph

3.2.1. Construction of dynamic association graph

To solve the problem that the fixed graph cannot follow the changes in association among the features, this paper proposes a dynamic association graph to represent such changes. As a widely used data relationship extraction method, Spearman rank correla-

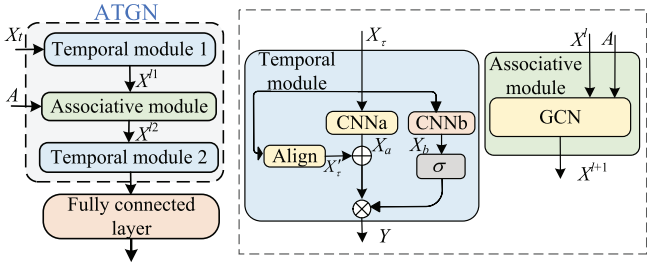

Fig. 3. The structure of ATGN.

tion is used to measure the associations among the features. In Eq. (3), ρ_{x_i,x_j} represents the Spearman rank correlation coefficient of variables x_i and x_j which are the data of different features. The $\mathrm{COV}(\cdot)$ represents the covariance operation, and $\mathrm{STD}(\cdot)$ is the standard deviation operation. $\mathrm{R}(x_i)$ and $\mathrm{R}(x_j)$ represent the corresponding ranks of x_i and x_j, respectively. The correlation coefficients between the features are modified according to Eq. (4), where $\epsilon$is the lowest correlation threshold, and $\epsilon = 0$in this paper. $A_r(X)$is the adjacency matrix got from data set X.

$$\rho_{x_i,x_j} = \frac{\mathrm{COV}(\mathrm{R}(x_i), \mathrm{R}(x_j))}{\mathrm{STD}(\mathrm{R}(x_i))\mathrm{STD}(\mathrm{R}(x_j))} \tag{3}$$

$$w_{ij} = \begin{cases} \left|\rho_{x_i,x_j}\right|, & i \neq j \text{and} \left|\rho_{x_i,x_j}\right| \geqslant \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$A_r(X) = \begin{bmatrix} 0 & w_{1,2} & \cdots & w_{1,n-1} & w_{1,n} \\ w_{2,1} & 0 & \cdots & w_{2,n-1} & w_{2,n} \\ w_{3,1} & w_{3,2} & \ddots & w_{3,n-1} & w_{3,n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n-1} & 0 \end{bmatrix} \tag{5}$$

To enable the structure to be updated with real-time data while retaining historical information, the proposed dynamic association graph G_{dyn} is constructed by two graphs, stable association graph G_{stable} and time-varying association graph G_{vary}, respectively. The construction methods of these graphs are described as follows:

(a) **Stable association graph:** As shown in Fig. 4, the historical data X_{stable} containing a large number of samples is used to construct the stable adjacency matrix A_{stable} of G_{stable} according to Eq. (6). In practice, the historical data is all available samples in the training set, so G_{stable} is fixed and reflects the overall associations among features.

(b) **Time-varying association graph:** To describe the latest associations among the features with the update of the real-time data, G_{vary} is constructed by using the latest data X_{vary} which only contains recent L samples. The adjacency matrix A_{vary} of G_{vary} can be constructed according to Eq. (7).

$$A_{stable} = A_r(X_{stable}) \tag{6}$$

$$A_{vary} = A_r(X_{vary}) \tag{7}$$

Fig. 4 shows the construction process of dynamic adjacency matrix A_{dyn} which represents the G_{dyn}. The dynamic association graph is proposed by combining the advantages of the G_{vary} and the G_{stable}. According to Eq. (8), the A_{dyn} is composed of the stable part A_{stable} and the dynamic part $A_{stable} \odot A_{vary}$, which are used to ensure the stability and the flexibility of A_{dyn}, respectively. It is worth noting that both parts contain A_{stable}, which ensures that A_{dyn} always contains the overall association information. The A_{one} is a $n \times n$ matrix whose element values are all 1. The steady-state coefficient matrix Θ is a $n \times n$ matrix whose element values are limited between 0 to 1, which can control the combination ratio of the stable and dynamic part, and limits the element values of the A_{dyn} from 0 to 1. It is worth noting that Θ can be defined as a learnable parameter to participate in the gradient descent during the training of the model.

$$A_{dyn} = \Theta \odot A_{stable} + (A_{one} - \Theta) \odot A_{stable} \odot A_{vary} \tag{8}$$

The dynamic association graph makes it possible for the model to extract the latest data association information reasonably. The steady-state coefficient matrix Θ limits the elements of A_{dyn} to a more reasonable range, and its trainability ensures the dynamic association graph can be reasonably applied to the CL prediction.

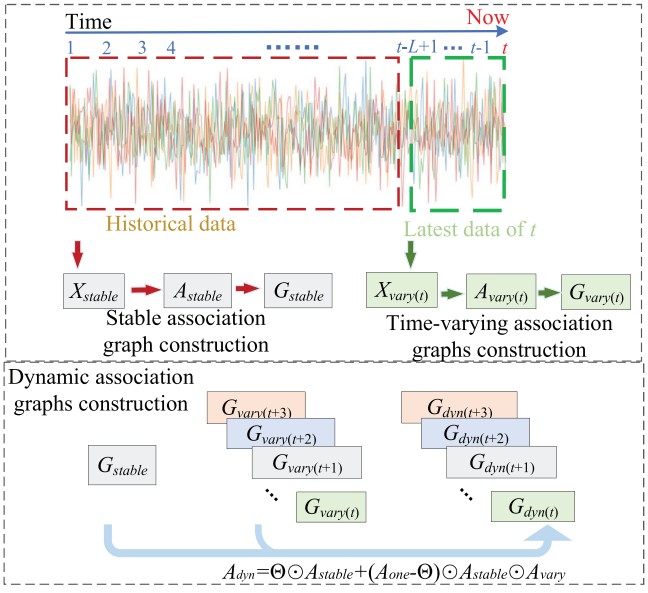

Fig. 4. Construction of stable association graph, time-varying association graph and dynamic association graph.

Algorithm 1: Empirical knowledge based feature classification

Input: The features that affect the CL prediction $F = \{f_1, f_2, f_3 \ldots f_n\}$, the f_n represents the nth feature; Classification guidelines $P1, P2, P3$.

Output: The feature category set $S_{KM} = \{s_1, s_2, s_3 \ldots s_m\}$;

```
P1: Both features represent a specific identical physical change;
P2: Both features are indicators obtained by the same calculation method;
P3: Both features represent time.
Initialize m = 1, s_1 = { }
for i = 1 to n − 1 do
    if f_i ∉ s_k(k = 1 ∼ m) then
        Classify f_i into feature category s_m
        for j = i + 1 to n do
            if f_i, f_j meet P1 or P2 or P3 then
                Classify f_j into feature category s_m
        if There are still unclassified features then
            m = m + 1
            s_m = { }
if f_n ∉ s_k(k = 1 ∼ m) then
    Classify f_n into feature category s_m
```

Result: The feature category set $S_{KM} = \{s_1, s_2, \ldots, s_m\}$

Algorithm 2: Knowledge embedded weighted mask construction

Input: The feature category set S_{KM}; the value of the maintain weight; the value of the adjust weight;

Output: Knowledge embedded weighted mask $A_{mask} \in \mathbb{R}^{n\times n}$.

According to $S_{KM} = \{s_1, s_2, s_3 \dots s_m\}$, the SL_m represents the number of features in the mth category s_m.
Initialize $row = 0$
for $k = 1$ *to* m **do**
  for $i = (row + 1)$ *to* $(row + SL_k)$ **do**
    for $j = 1$ *to* n **do**
      if $j \in [row + 1, row + SL_k]$ **then**
        $A_{mask}(i, j) =$ maintain weight
      else
        $A_{mask}(i, j) =$ adjust weight
  $row = row + SL_k$

Result: Knowledge embedded weighted mask A_{mask}

3.2.2. Knowledge embedding method

As mentioned above, the types of the features are various. Their data have certain differences in the order of magnitude and unit. The characteristics of the features may be fuzzified in the aggregation process of GCN, which may reduce the diversity of information, thereby affecting the feature extraction quality of the model. This paper proposes a knowledge embedded weighted mask (KM) mechanism to embed the empirical knowledge into the graph. By classifying the features into different categories and adjusting the associations according to the empirical knowledge, the KM can protect the data diversity from the over-aggregation of the features. Based on the KM, the knowledge embedded dynamic association graph is constructed, as shown in Fig. 5, whose process is described as follows:

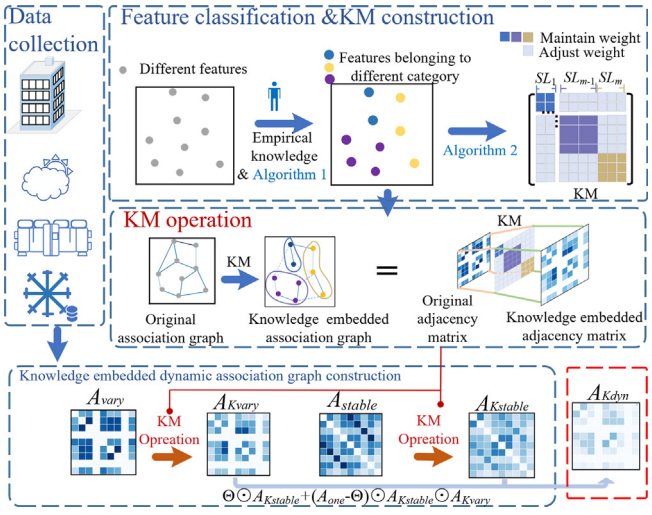

Fig. 5. Construction method of the knowledge embedded dynamic association graph.

(a) **Empirical knowledge based feature classification:** The features that affect the *CL* prediction can be manually divided into $m \in [1, n]$ categories according to Algorithm 1. The $P1, P2$ and $P3$ are the empirical knowledge, which serves as the guidelines for feature classification. Based on the classification, the feature category set $S_{KM} = \{s_1, s_2, s_3 \dots s_m\}$is obtained. The SL_m represents the number of features classified into mth feature category s_m. It is worth noting that such number may be different in the different categories, and the features belonging to the same category have similar characteristics.
(b) **KM construction:** Based on the S_{KM}, the KM matrix $A_{mask} \in \mathbb{R}^{n\times n}$ is constructed according to Algorithm 2, which contains adjust weight (AW) and maintain weight (MW). MW is usually set to 1 to maintain the associations among the features belonging to the same category. AW is real numbers between 0 and 1, and the selection of AW is based on empirical knowledge.
(c) **Knowledge embedded dynamic association graph construction:** KM operation is carried out according to Eqs. (9)~(10), and the associations among the features belonging to the same category are maintained, while those belonging to different categories are adjusted. The knowledge embedded stable adjacency matrix $A_{Kstable}$ and knowledge embedded time-varying adjacency matrix A_{Kvary} are obtained. These adjacency matrix are brought into Eq. (8) to get the knowledge embedded dynamic adjacency matrix A_{Kdyn}, which represents the knowledge embedded dynamic association graph.

$$A_{Kstable} = A_{stable} \odot A_{mask} \tag{9}$$

$$A_{Kvary} = A_{vary} \odot A_{mask} \tag{10}$$

Based on the KM mechanism, the empirical knowledge can be effectively embedded into the dynamic association graph, which guides a more rational information aggregation of the model.

3.3. The framework of proposed method

To integrate the dynamic associations and temporal information of the features, this paper proposes the associative-temporal dynamic graph convolutional network (ATDGN) by introducing the dynamic association graph into the ATGN, as shown in Fig. 6. When the knowledge is embedded into the graph, K-ATDGN is constructed. One of the critical parameters of the proposed method is the Θ, which may affect the result of the graph convolution and further influents the prediction effect. To find a suitable Θ, it is defined as a learnable parameter embedded in the ATDGN to par-

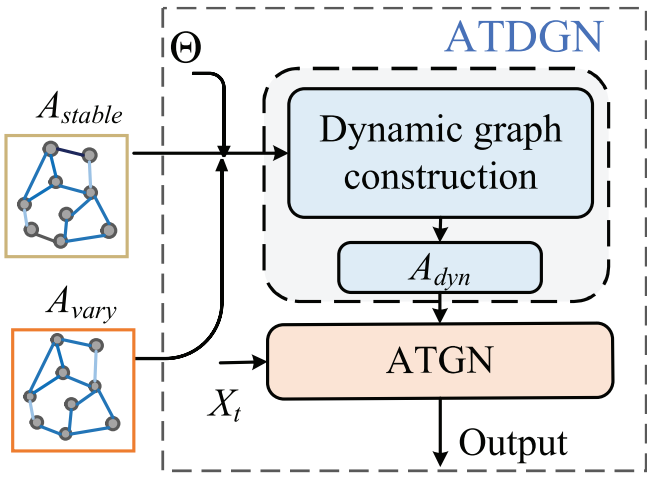

Fig. 6. The structure of ATDGN.

ticipate in the gradient descent. The framework of the proposed K-ATDGN is shown in Fig. 7, and a detailed description of each step is given below:

(a) **Step1:** Collect data of the features that affect *CL* prediction from the onsite database, then screen and preprocess the data preliminarily.
(b) **Step2:** According to Eqs. (6)∼(7), historical data and lastest data are used to construct the A_{stable} and A_{vary}, respectively;
(c) **Step3:** According to the empirical knowledge and Algorithm 1, the features are divided into several categories, and the KM is constructed according to Algorithm 2. Based on the KM, the $A_{K\,vary}$ and $A_{Kstable}$ are obtained referring to Eqs. (9)∼(10).
(d) **Step4:** According to Eq. (8), the $A_{K\,vary}$ and $A_{Kstable}$ are used to build the knowledge embedded dynamic association graph;
(e) **Step5:** The X^{l1}, X^{l2} and X^{l3} are obtained step by step through temporal module1, associative module and temporal module2. Finally, through the fully connected layer, the *CL* prediction value is obtained from X^{l3}.

4. Experimental study

4.1. Data collection and feature classification

The research data is collected from a building located in Shenzhen, Guangdong, China. The building covers an area of around 11200m^2. There are forty-five floors above the ground and four floors underground. The general structure of chiller plant of the building is shown in Fig. 8 and the scene of which is shown in Fig. 9. The plant includes a water collection tank, a water distribution tank, and four chillers. Each chiller is equipped with a cooling water pump and a chilled water pump. The water collection tank is responsible for collecting chilled water that has absorbed heat in the building and distributing the water to the chillers. The water distribution tank collects the cooled chilled water from the chillers and distributes it to the designated building areas. As the core

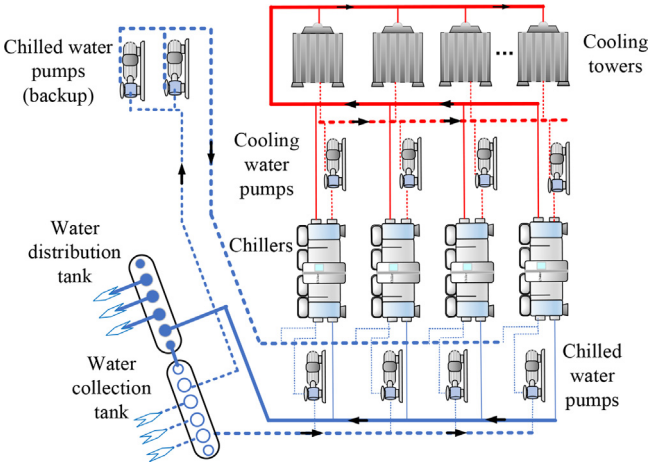

Fig. 8. The structure of the chiller plant.

Fig. 9. The scene of the chiller plant.

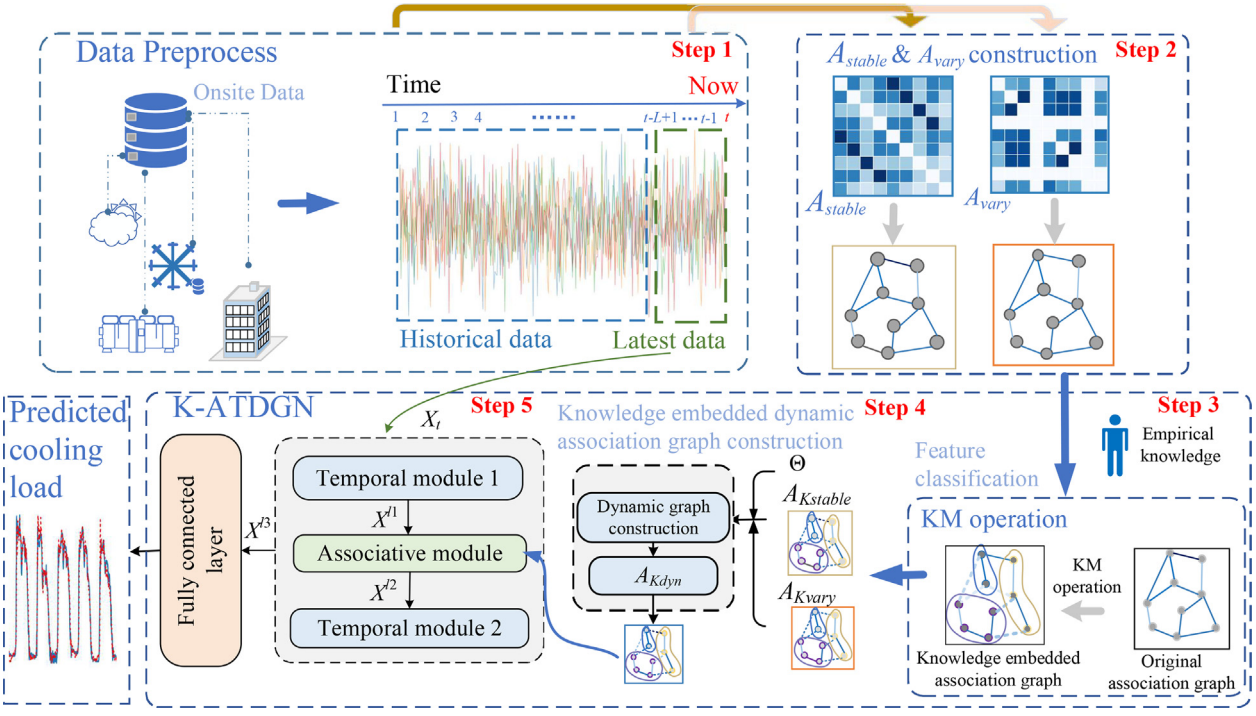

Fig. 7. The framework of the proposed method.

refrigeration equipment of the chiller plant, the chillers cooperate with the chilled water pumps, cooling water pumps, and outdoor cooling towers to complete most cooling operations.

This paper collects the operation data of key refrigeration equipment, the outdoor meteorological features, and the time features. All these features are obtained based on real-time data. In terms of meteorological features, this paper acquired four easily available meteorological data including outdoor dew point temperature, outdoor humidity, outdoor temperature and outdoor wet bulb temperature. Moreover, during the time period of the collected data, the area where the building is located has a variety of weather conditions, such as rainy, sunny, cloudy and overcast, which makes experimental data more abundant. Time features include week, day, hour and minute. Equipment operating status features include contains a lot of device information. Table 1 shows all the collected features. The features are classified into different feature categories according to Algorithm 1.

4.2. Data division and association graph construction

This paper collects 35136 samples from 00:00 on June 1st, 2021, to 23:55 on September 30th, 2021, with a sampling interval of 5 min. In each sample, 81 features of *CL* prediction are included. Fig. 10 shows the chilled water inlet and outlet temperature of two of four chillers from June 1st, 2021 to June 10th, 2021. The original collected data $X_{ori} \in \mathbb{R}^{1\times 35136\times 81}$ is divided, standardized, then undergo a sliding window operation: Firstly, the data was divided and standardized to get the training set X_{train}, validation set X_{val}, and test set X_{test}. Then, each data set is slid into many data blocks with the same number of samples. As shown in Fig. 11, in the sliding window operation, the window length L is 12, and the step size S is set to 2, thus each data block contains samples of one hour. The data forms after such an operation are provided in Table 2. This paper uses X_{train} to construct the stable association graph. The time-varying association graphs are constructed according to the data after window sliding. In other words, the stable association graph is constructed according to months of data, while the time-varying association graphs are built according to the data of the recent hour before the predicted time point.

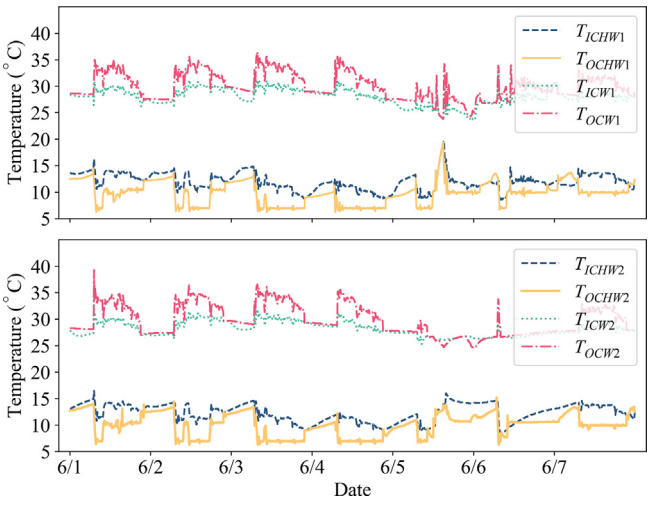

Fig. 10. The cooling water and chilled water temperature.

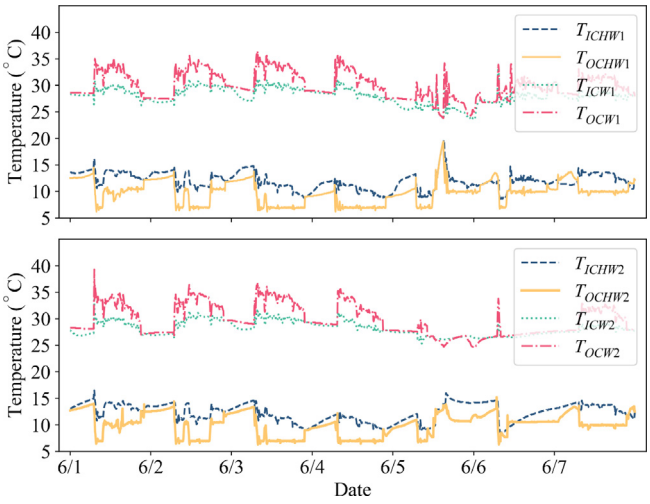

Fig. 11. The example of silding window operation and graph construction method.

Table 1
Feature classification.

	Feature name				Feature introduction	Feature category	SL_M
Time	t_{week}	t_{day}	t_{hour}	t_{min}	Week; Day; Hour; Minute	s_1	4
Equipment operating status features	Q_1	Q_2	Q_3	Q_4	Cooling load borne by four chillers	s_2	4
	PW_1	PW_2	PW_3	PW_4	Power of four chillers	s_3	4
	COP_1	COP_2	COP_3	COP_4	Coefficient of performance of four chillers	s_4	4
	$Flow_1$	$Flow_2$	$Flow_3$	$Flow_4$	Flow rate of four chillers' chilled water	s_5	4
	T_{ICHW1}	T_{ICHW2}	T_{ICHW3}	T_{ICHW4}	Temperature of four chillers' inlet chilled water	s_6	4
	T_{OCHW1}	T_{OCHW2}	T_{OCHW3}	T_{OCHW4}	Temperature of four chillers' outlet chilled water	s_7	4
	T_{OCW1}	T_{OCW2}	T_{OCW3}	T_{OCW4}	Temperature of four chillers' outlet cooling water	s_8	4
	T_{ICW1}	T_{ICW2}	T_{ICW3}	T_{ICW4}	Temperature of four chillers' inlet cooling water	s_9	4
	T_{CA1}	T_{CA2}	T_{CA3}	T_{CA4}	Condensation approach temperature of four chillers	s_{10}	4
	T_{EA1}	T_{EA2}	T_{EA3}	T_{EA4}	Evaporation approach temperature of four chillers	s_{11}	4
	P_{C1}	P_{C2}	P_{C3}	P_{C4}	Condensation pressure of four chillers	s_{12}	4
	P_{E1}	P_{E2}	P_{E3}	P_{E4}	Evaporation pressure of four chillers	s_{13}	4
	F_{CW1}	F_{CW2}	F_{CW3}	F_{CW4}	Frequency of four cooling water pumps	s_{14}	4
	F_{CHW1}	F_{CHW2}	F_{CHW4}	F_{CHW4}	Frequency of four chilled water pumps	s_{15}	4
	F_{TW1}	F_{TW2} ...	F_{TW15}	F_{TW16}	Frequency of sixteen cooling towers	s_{16}	16
	Q_{total}				Total cooling load	s_{17}	1
Meteorological features	T_{DP}				Outdoor dew point temperature	s_{18}	1
	H_O				Outdoor humidity	s_{19}	1
	T_O				Outdoor temperature	s_{20}	1
	T_{WB}				Outdoor wet bulb temperature	s_{21}	1

Table 2
Dataset division details.

Dataset classification	Time span	Data form after standardization	Data form after sliding window operation
Traing set	00:00, Jun. 1st, 2021 ~23:55, Aug. 10th, 2021	$X_{train} \in \mathbb{R}^{1\times 20448\times 81}$	$X_{W-train} \in \mathbb{R}^{10219\times 1\times 12\times 81}$
Validation set	00:00, Aug. 11st, 2021 ~23:55, Sept. 10th, 2021	$X_{val} \in \mathbb{R}^{1\times 8928\times 81}$	$X_{W-val} \in \mathbb{R}^{4459\times 1\times 12\times 81}$
Testing set	00:00, Sept. 11st, 2021 ~23:55, Sept. 30th, 2021	$X_{test} \in \mathbb{R}^{1\times 5760\times 81}$	$X_{W-test} \in \mathbb{R}^{2875\times 1\times 12\times 81}$

4.3. Evaluation indicators

The mean absolute percentage error (MAPE), the root mean squared error (RMSE), the coefficient of determination R^2 and coefficient of variation of the root mean square error (CVRMSE) between the prediction results of the model and the actual value are taken as the evaluation indicators. All these evaluation indicators can be calculated according to Eqs. (11)~(15). In these equations, $\hat{y}_i, y_i, n$ represent the ith predicted value, the ith actual value and the number of samples, respectively. When $y_i = 0, y_i$ and the corresponding $\hat{y}_i$ will not participate in the MAPE calculation.

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{11}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{12}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{13}$$

$$\mathrm{CVRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}}{\bar{y}} \tag{14}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{15}$$

4.4. Results and discussion

4.4.1. Discussion on the effect of dynamic association graph on cooling load prediction

Comparative experiments are conducted in this subsection to show the effectiveness of the dynamic association graph. ATGN based on the stable association graph, the time-varying association graph and dynamic association graph (abbreviated as $G_{vary}, G_{stable}, G_{dyn}$, respectively) are compared in 15-min ahead and 30-min ahead prediction tasks.

It is worth mentioning that the output channels of temporal module1, associative module and temporal module2 of ATGN are set to 32, 64 and 1, respectively. The kernel sizes of the temporal modules are all set to 3. Adam optimizer is used in the training of the models, and the learning rate is kept at 0.001. The maximum training epoch is 600. The training set is used to train the model, and the model under the corresponding epoch with the minimum MAPE in the validation set is selected as the optimal model during training. Five repeated experiments are performed to ensure the results' reliability. The results on the test set are given in Table 3.

The experimental results show that the proposed G_{dyn} based ATGN (ATDGN) has a better *CL* prediction effect compared with the G_{vary} based one and the G_{stable} based one. In the 15-min ahead *CL* prediction, the average MAPE of the ATDGN is 16.94%, which is lower than the G_{vary} based ATGN (23.04%) and the G_{stable} based one (17.86%). The RMSE, R^2 and CVRMSE of the ATDGN are 398.68 kW, 0.9652 and 14.12% respectively, all of which are in an acceptable range. In the 30-min ahead *CL* prediction, the MAPE, RMSE, R^2 and CVRMSE of the G_{dyn} based ATGN are 21.19%, 518.66 kW, 0.9413 and 18.38% respectively. All the indicators of G_{dyn} based ATGN perform better than the G_{stable} based one and the G_{vary} based one.

It can conclude from the experimental results that the limitation of the sample number and the time-varying of the data lead to the instability of the G_{vary}, so the accuracy of ATGN based on such a graph is limited. When the G_{stable} is used to characterize the associations among the features, which is constructed according to a large amount of data, the ATGN can meet an ideal prediction accuracy. The dynamic association graph combines the advantages of the G_{vary} and G_{stable}, and by introducing the steady-state coefficient matrix, such an association graph can adjust the graph structure during the training of the model. Thus, the G_{dyn} based ATGN achieves a superior predicted result, which shows the effect of the dynamic association graph on the *CL* prediction.

Table 3
Prediction performances of ATGNs based on different association graphs.

Time Span		15-min			30-min		
Association Graph	Indicator	Average	Max	Min	Average	Max	Min
G_{stable}	MAPE	17.86	20.00	16.27	21.78	24.60	20.92
	RMSE	401.52	434.02	379.41	544.88	569.77	529.79
	R^2	0.9648	0.9686	0.9589	0.9352	0.9388	0.9292
	CVRMSE	14.22	15.38	13.44	19.30	20.19	18.77
G_{vary}	MAPE	23.04	23.57	22.74	28.47	29.95	26.55
	RMSE	448.60	457.71	438.76	598.26	612.93	590.75
	R^2	0.9561	0.9580	0.9543	0.9220	0.9239	0.9181
	CVRMSE	15.89	16.22	15.54	21.20	21.72	20.93
G_{dyn}	MAPE	**16.94**	**17.78**	**16.01**	**21.19**	**21.56**	**20.44**
	RMSE	**398.68**	**430.38**	**370.66**	**518.66**	**543.17**	**501.06**
	R^2	**0.9652**	**0.9700**	**0.9596**	**0.9413**	**0.9453**	**0.9357**
	CVRMSE	**14.12**	**15.25**	**13.13**	**18.38**	**19.24**	**17.75**

Bold font represents best results; the units of MAPE and CVRMSE are %, and the unit of RMSE is kW.

Table 4
Prediction performances of K-ATDGNs with different adjust weight.

Time Span		15-min			30-min		
Adjust weight	Indicator	Average	Max	Min	Average	Max	Min
0.2	MAPE	15.10*	**16.13**	14.18*	20.95	22.31	19.66
	RMSE	342.37*	366.11*	327.82*	505.29	552.92	482.43
	R^2	0.9744*	0.9766*	0.9708*	0.9442	0.9493	0.9334
	CVRMSE	12.13*	12.97*	11.61*	17.90	19.59	17.09
0.4	MAPE	**14.92**	16.48*	**14.14**	20.60*	22.40	**19.16**
	RMSE	**340.73**	**352.54**	**323.29**	**484.59**	**494.54**	**477.95**
	R^2	**0.9747**	**0.9772**	**0.9729**	**0.9488**	**0.95021**	**0.9467**
	CVRMSE	**12.07**	**12.49**	**11.45**	**17.17**	**17.52**	**16.93**
0.6	MAPE	15.70	17.00	14.95	**20.54**	23.14	19.20*
	RMSE	364.61	406.30	342.79	496.26*	504.20*	478.08*
	R^2	0.9709	0.9744	0.9640	0.9463*	0.95018*	0.9446*
	CVRMSE	12.92	14.39	12.14	17.58*	17.86*	16.94*
0.8	MAPE	16.10	16.54	15.50	20.99	21.92*	19.88
	RMSE	378.68	431.89	356.18	529.76	547.43	519.39
	R^2	0.9686	0.9723	0.9593	0.9388	0.9412	0.9347
	CVRMSE	13.42	15.30	12.62	18.77	19.39	18.40
1	MAPE	16.94	17.78	16.01	21.19	**21.56**	20.44
	RMSE	398.68	430.38	370.66	518.66	543.17	501.06
	R^2	0.9652	0.9700	0.9596	0.9413	0.9453	0.9357
	CVRMSE	14.12	15.25	13.13	18.38	19.24	17.75

Bold font represents best results, * represents the second best results; the units of MAPE and CVRMSE are %, and the unit of RMSE is kW.

4.4.2. Discussion on the effect of knowledge embedding on cooling load prediction

Based on the empirical knowledge, the features are classified into several categories. The adjust weight (AW) determines the adjust degree of the associations among the features belonging to different categories. When the value of AW is 1, during the process of GCN, information is aggregated according to the dynamic graph G_{dyn} without embedded knowledge. When AW is less than 1, the associations is adjusted. The influence of AW on the *CL* prediction accuracy is discussed in this subsection, and the effect of knowledge embedded dynamic association graphs based ATDGNs (K-ATDGNs) are compared with other graphs based ATGNs. K-ATDGNs are trained under five different AWs, whose values are 0.2, 0.4, 0.6, 0.8, and 1, respectively. The 15-min ahead prediction and 30-min ahead predictions are carried out. The training strategies of the models are the same as that described in subSection 4.4.1, and the experimental results on the test set are given in Table 4. The experimental results under different prediction time spans are shown in Fig. 12.

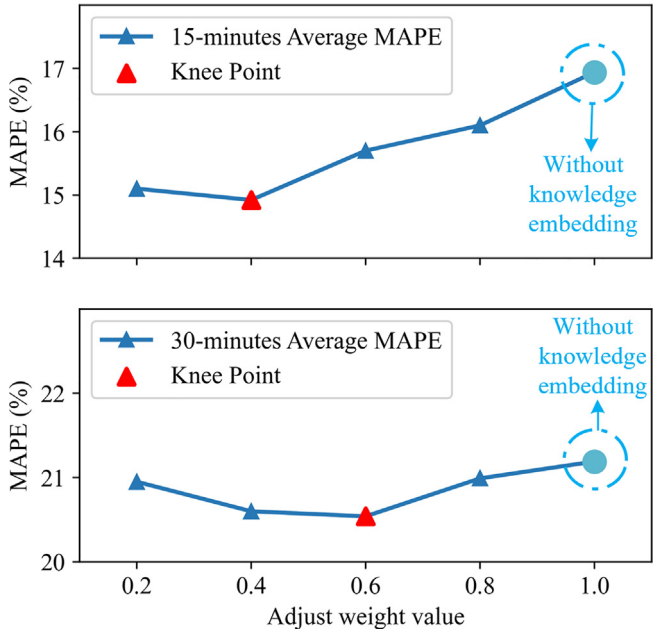

Fig. 12. K-ATDGN prediction performance under different adjust weights in 15-min ahead prediction and 30-min ahead prediction.

It can be seen that in 15-min ahead prediction, when AW = 0.4, the average MAPE, RMSE, R^2 and CVRMSE of K-ATDGN are relatively better, which are 14.92%, 340.73 kW, 0.9747 and 12.07% respectively. When AW > 0.4, the average MAPE, RMSE and CVRMSE gradually increase, and the average R^2 decreases gradually. When the empirical knowledge is not embedded, the average MAPE, RMSE, R^2 and CVRMSE are 16.94%, 398.68 kW, 0.9652 and 14.12%, respectively. In the 30-min ahead *CL* prediction, the knee point of the average MAPE occurs when AW = 0.6, which is 20.54%. The average RMSE is 496.26 kW, R^2 is 0.9463 and CVRMSE is 17.58% when AW = 0.6, both of which are in an acceptable range.

Combining the experimental results of this subsection and subSection 4.4.1, it can be concluded that the graph structures are more reasonable due to the adjustment of the associations, which is guided by empirical knowledge. The MAPEs of ATGNs based on different association graphs are shown in Fig. 13. Whether in 15-min ahead prediction or 30-min ahead prediction, the average RMSE, MAPE, R^2 and CVRMSE of the K-ATDGN with a

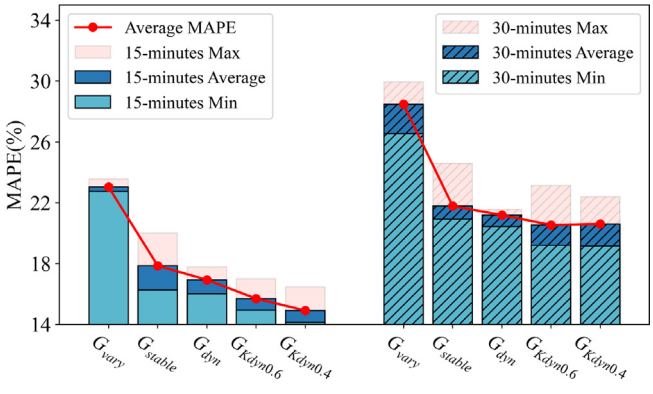

Fig. 13. MAPEs of ATGNs based on different association graphs.

suitable AW are in the reasonable range, which are superior to the other graphs based ATGNs mentioned above (Knowledge embedded dynamic association graph with AW = 0.4 and AW = 0.6 are abbreviated as $G_{Kdyn0.4}$ and $G_{Kdyn0.6}$, respectively). Therefore, when knowledge is reasonably embedded into the graph, it can positively impact the *CL* prediction.

The adjustability of AW brings more flexibility to the association graph, and it can be seen that compared to keeping the original associations among the features, embedding knowledge to adjust the association graph is an effective approach to improve the prediction accuracy.

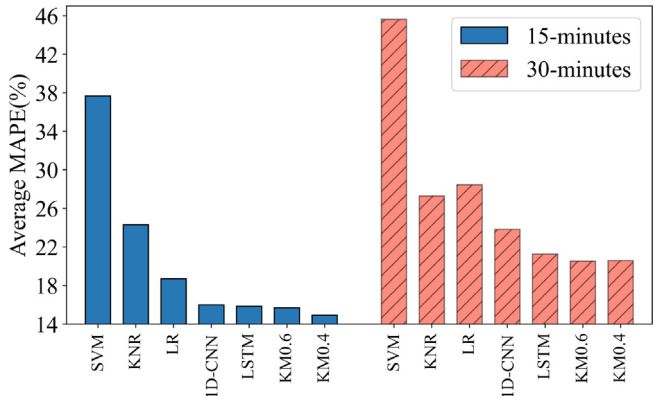

Fig. 14. The prediction performance of different methods. (The KM0.4 and KM0.6 represent the K-ATDGN with AW = 0.4 and AW = 0.6, respectively).

4.4.3. Comparison with other methods

To verify the effectiveness of the K-ATDGN, this paper compares K-ATDGN with one dimension CNN (1D-CNN), LSTM. Some traditional machine learning algorithms with easily determined hyperparameters and thus easy to deploy on site, such as SVM, LinearRegression (LR) and KNeighbors Regressor (KNR), are also used for comparison, too. The K-ATDGN, 1D-CNN and LSTM are implemented by Pytorch [24]. The optimizer, learning rate and optimal model acquisition method used in training are the same as is used in subSection 4.4.1. Among them, 1D-CNN uses four convolutional layers and one fully connected layer, the channels of output data of each convolutional layer are 32, 64, 32 and 1, respectively. The kernel size of each convolutional layer is set to 2. The LSTM has a hidden layer of which dimension is set to 512 and a fully connected layer. In the K-ATDGN related experiments, AW is taken as 0.4 and 0.6. The experiments of 1D-CNN, LSTM and K-ATDGN repeat five times.

The SVM, LR and KNR are implemented by Scikit-learn [25]. SVM kernel is set as RBF, the kernel coefficient for RBF *gamma* is set to 0.001, and the regulation parameter is 50. Regarding KNR, the number of neighbors is set to 10, and the power parameter for the Minkowski metric is set to 1. The intercept is taken into account in the LR. It should be noted that the predicted results of these three methods do not change in the repeated experiments.

From Fig. 14 and Table 5, it can be seen that, compared with other methods provided, the K-ATDGNs have excellent prediction performance in both 15-min and 30-min ahead prediction: in 15-min ahead prediction, the average MAPE of the proposed K-ATDGN (AW = 0.4) are 14.92%, which is 15.70% when AW = 0.6. The average RMSE of K-ATDGN (AW = 0.4) and K-ATDGN (AW = 0.6) are 340.73 kW and 364.61 kW, respectively, while the RMSE of other methods is bigger. Compared with KNR, the RMSE of the K-ATDGN (AW = 0.4) is reduced by about 43%. The average

Table 5
Family Comparison of different models in cooling load prediction.

Time Span		15-min			30-min		
Model	Indicator	Average	Max	Min	Average	Max	Min
K-ATDGN (AW = 0.4)	MAPE	**14.92**	**16.48**	**14.14**	20.60*	**22.40**	**19.16**
	RMSE	**340.73**	**352.54**	**323.29**	**484.59**	**494.54**	**477.95**
	R^2	**0.9747**	**0.9772**	**0.9729**	**0.9488**	**0.95021**	**0.9467**
	CVRMSE	**12.07**	**12.49**	**11.45**	**17.17**	**17.52**	**16.93**
K-ATDGN (AW = 0.6)	MAPE	15.70*	17.00	14.95	**20.54**	23.14	19.20*
	RMSE	364.61*	406.30	342.79*	496.26*	504.20*	478.08*
	R^2	0.9709*	0.9744*	0.9640	0.9463*	0.95018*	0.9446*
	CVRMSE	12.92*	14.39	12.14*	17.58*	17.86*	16.94*
LSTM	MAPE	15.85	17.20	14.75*	21.26	22.45*	19.65
	RMSE	379.33	395.72	370.57	544.91	559.97	524.58
	R^2	0.9686	0.9701	0.9659	0.9352	0.9400	0.9317
	CVRMSE	13.44	14.02	13.13	19.31	19.84	18.59
1D-CNN	MAPE	16.00	16.67*	14.93	23.84	25.45	21.66
	RMSE	367.37	385.67*	355.65	522.85	532.08	512.67
	R^2	0.9705	0.9724	0.9676*	0.9404	0.9427	0.9383
	CVRMSE	13.01	13.66*	12.60	18.52	18.85	18.16
LR	MAPE	18.72	18.72	18.72	28.45	28.45	28.45
	RMSE	398.17	398.17	398.17	573.80	573.80	573.80
	R^2	0.9654	0.9654	0.9654	0.9282	0.9282	0.9282
	CVRMSE	14.11	14.11	14.11	20.33	20.33	20.33
KNR	MAPE	24.33	24.33	24.33	27.29	27.29	27.29
	RMSE	598.97	598.97	598.97	717.00	717.00	717.00
	R^2	0.9218	0.9218	0.9218	0.8880	0.8880	0.8880
	CVRMSE	21.22	21.22	21.22	25.40	25.40	25.40
SVM	MAPE	37.67	37.67	37.67	45.63	45.63	45.63
	RMSE	540.48	540.48	540.48	699.29	699.29	699.29
	R^2	0.9363	0.9363	0.9363	0.8934	0.8934	0.8934
	CVRMSE	19.15	19.15	19.15	24.77	24.77	24.77

Bold font represents best results, * represents the second best results; the units of MAPE and CVRMSE are %, and the unit of RMSE is kW.

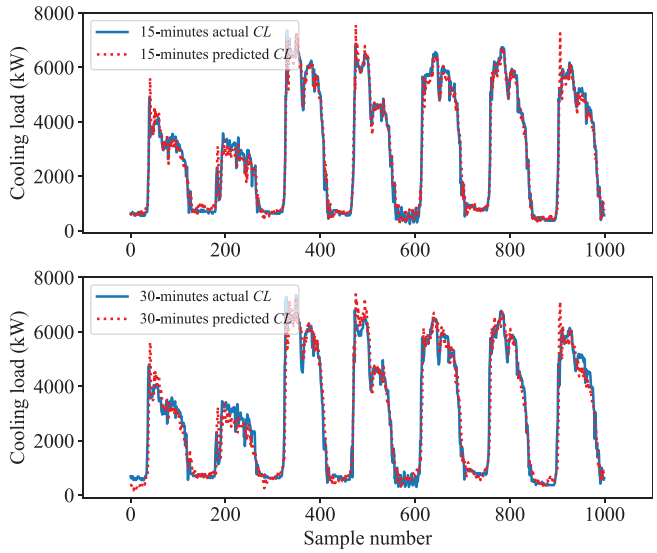

Fig. 15. Comparison between the actual *CL* and the K-ATDGN predicted *CL*.

CVRMSE of K-ATDGN (AW = 0.4) and K-ATDGN (AW = 0.6) are 12.07% and 12.92%, respectively. In addition, the average R^2 of the K-ATDGN (AW = 0.4) and the K-ATDGN (AW = 0.6) are also greater than the R^2 of other prediction methods. In the 30-min ahead prediction, K-ATDGN still performs well whether with AW = 0.4 or AW = 0.6. The average MAPE of K-ATDGN is 20.54% when AW = 0.6, which is lower than that of other prediction methods. In addition, the average RMSE, R^2 and CVRMSE of the K-ATDGNs are better than other methods.

The performance of K-ATDGN in the partial test set is shown in Fig. 15. It can be seen that the proposed model can accurately capture the trend of the *CL*, and the prediction effect is excellent. The proposed K-ATDGN is able to combine the temporal information, associations among the features and empirical knowledge to obtain an accurate predicted result, which has practical value in the field of *CL* prediction.

5. Conclusion

This paper proposes a new *CL* prediction method which integrates the association and temporal domain information of the features that affect the *CL* prediction. To flexibly reflect the time-varying change of the associations among the features, the proposed method combines historical data with the latest data to construct a dynamic association graph, and these data provides more information for the graph structure. Further, the dynamic graph can adjust the structure during the model training, which improves the adaptability of the method. By analyzing the prediction effects based on different association graphs, the experimental results show that the proposed dynamic association graph positively impacts the prediction accuracy.

In addition, a knowledge embedding method is proposed to adjust the associations among the features, which provides a reasonable and easy way to quantify the empirical knowledge and aims to ensure the rationality of such associations and keep the diversity of information. The experimental results show that the introduction of empirical knowledge can guide the model to extract implicit information more effectively as a way to improve the prediction accuracy.

The comprehensive comparison experiments with traditional prediction methods are carried out on the challenging onsite data, and the *CL* prediction accuracy of the proposed K-ATDGN is better than that of other methods. The experiments show the combination of temporal information and dynamic data associations, coupled with the introduction of empirical knowledge, can significantly improve the accuracy of *CL* prediction.

The further research designed to build on the findings of this study: since the number and importance of features will affect the calculation efficiency and prediction accuracy of the model, to achieve a reasonable balance between features number and prediction accuracy, a method and a quantifiable indicator are required to guide the selection of features while ensuring the accuracy of prediction. The associations between the features directly affect the construction of the graph, so it is also the focus of future research to explore the relationship between nodes using different metrics. In addition, integrating more knowledge, such as the importance of features to cooling load prediction, into the construction of the graph is worth more investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the Science and Technology Innovation Program of Hunan Province (#2022RC1090, #2021RC4054),in part by the National Natural Science Foundation of China (#62173349,#61988101),in part by the National Natural Science Foundation of Hunan Province (#2022JJ20076).

References

[1] N. Somu, G.R. MR, K. Ramamritham, A deep learning framework for building energy consumption forecast, Renew. Sustain. Energy Rev. 137 (2021) 110591.
[2] A. Costa, M.M. Keane, J.I. Torrens, E. Corry, Building operation and energy performance: Monitoring, analysis and optimisation toolkit, Appl. Energy 101 (2013) 310–316.
[3] C. Fan, F. Xiao, Y. Zhao, A short-term building cooling load prediction method using deep learning algorithms, Appl. Energy 195 (2017) 222–233.
[4] O.M. Al-Rabghi, K.M. Al-Johani, Utilizing transfer function method for hourly cooling load calculations, Energy Convers. Manage. 38 (4) (1997) 319–332.
[5] G.J. Schoenau, R.A. Kehrig, A correlation method for predicting monthly and annual cooling loads in direct gain passive solar heated buildings, Energy Convers. Manage. 29 (3) (1989) 175–187.
[6] S.F. Bruning, A new way to calculate cooling loads, Ashrae J. 46 (1) (2004) 20.
[7] J. Deng, P. Jirutitijaroen, Short-term load forecasting using time series analysis: A case study for Singapore, in: 2010 IEEE Conference on Cybernetics and Intelligent Systems, IEEE, 2010, pp. 231–236.
[8] Y. Guo, E. Nazarian, J. Ko, K. Rajurkar, Hourly cooling load forecasting using time-indexed ARX models with two-stage weighted least squares regression, Energy Convers. Manage. 80 (2014) 46–53.
[9] A. Vaghefi, M.A. Jafari, E. Bisse, Y. Lu, J. Brouwer, Modeling and forecasting of cooling and electricity load demand, Appl. Energy 136 (2014) 186–196.
[10] Z. Hou, Z. Lian, An application of support vector machines in cooling load prediction, in: 2009 International Workshop on Intelligent Systems and Applications, IEEE, 2009, pp. 1–4.
[11] C. Fan, Y. Liao, G. Zhou, X. Zhou, Y. Ding, Improving cooling load prediction reliability for HVAC system using Monte-Carlo simulation to deal with uncertainties in input variables, Energy Build. 226 (2020).
[12] S. Idowu, S. Saguna, C. Åhlund, O. Schelén, Applied machine learning: Forecasting heat load in district heating system, Energy Build. 133 (2016) 478–488.
[13] A.E. Ben-Nakhi, M.A. Mahmoud, Cooling load prediction for buildings using general regression neural networks, Energy Convers. Manage. 45 (13–14) (2004) 2127–2141.
[14] Y.I. Alamin, J.D. Álvarez, M. del Mar Castilla, A. Ruano, An Artificial Neural Network (ANN) model to predict the electric load profile for an HVAC system, IFAC-PapersOnLine 51 (10) (2018) 26–31.
[15] Y. Yao, Z. Lian, Z. Hou, W. Liu, An innovative air-conditioning load forecasting model based on RBF neural network and combined residual error correction, Int. J. Refrig. 29 (4) (2006) 528–538.
[16] L. Wang, E.W. Lee, R.K. Yuen, Novel dynamic forecasting model for building cooling loads combining an artificial neural network and an ensemble approach, Appl. Energy 228 (2018) 1740–1753.

[17] J. Yu, Q. Liu, A. Zhao, X. Qian, R. Zhang, Optimal chiller loading in HVAC System Using a Novel Algorithm Based on the distributed framework, J. Build. Eng. 28 (2020).
[18] R. Sendra-Arranz, A. Gutiérrez, A long short-term memory artificial neural network to predict daily HVAC consumption in buildings, Energy Build. 216 (2020).
[19] A. Li, F. Xiao, C. Zhang, C. Fan, Attention-based interpretable neural network for building cooling load prediction, Appl. Energy 299 (2021).
[20] Z. Wang, T. Hong, M.A. Piette, Building thermal load prediction through shallow machine learning and deep learning, Appl. Energy 263 (2020).
[21] G. Qiang, T. Zhe, D. Yan, Z. Neng, An improved office building cooling load prediction model based on multivariable linear regression, Energy Build. 107 (2015) 445–455.
[22] T.N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: International Conference on Learning Representations (ICLR), 2017.
[23] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, in: IJCAI, 2018.
[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch.
[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.